Intelligent Crash Detection and Emergency Alert Mechanism for Autonomous Vehicles Using CrashGuard X Technology

S M Kowsik Godavarthi[1], Komal Sai Raj Atmakuri[2], Venkata Praveen Akula[3], Tanmayi Movva[4], Sai Revanth Singavarapu[5], Shrinath Munagapati[6], Srikanth Annamareddy[7]

[1,4,5,6,7] Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
[2]Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
[3] Department of ECS, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India

Emails: godavarthikowsik@gmail.com komalsairaj012@gmail.com akulapraveen535@gmail.com tanmayimovva30@gmail.com revanth.2074@gmail.com shrinath.munagapati@gmail.com annamareddy.srikanth123@kluniversity.in

---***---

Abstract - The safety features of self-driving cars call for ever greater innovation with regard to crash awareness and notification technologies. In this paper, we develop an Intelligent Crash Detection and Emergency Alert for self-driving cars that is based on CrashGuard X Technology with hardware integration of MPU6050, ESP32, and Arduino IDE for accident detection in real time. The system employs an acceleration and rotation monitoring device, the MPU6050, to track sudden variations of impact. These impacts are measured and processed by a microcontroller, an ESP32, which is programmed with Embedded C to make decisions. As soon as the device processes a crash situation, the ESP32 sends an SOS alarm over a Wi-Fi network, providing an additional advantage of being able to communicate with other devices and informing them of the emergency situation. The overall objective remains which is to make the device lightweight, to consume low energy, and to be useful for future applications of self-driving cars. The tests were conducted which showed highly accurate crash detection with a limited number of false alerts improving the overall response time. The proposed solution is an increase in safety while responding to accidents, the capability of the vehicle to operate autonomously in a reliable manner, an important step forward in intelligent transportation systems and advanced drivers' assistance systems.

Key Words: Crash Detection, Emergency Alert, Autonomous Vehicles, ESP32, MPU6050

1. INTRODUCTION

The advent of autonomous vehicles has begun with a vivid transformation of the transport system. There, however, arises the challenge of providing a safe, smart system for the protection of the passengers in conjunction with the protection of others in the road environment. Most old-fashioned crash-detection systems involve some external sensors and/or manual inputs which can lead to delays in the detection of collisions, and thus in activating appropriate emergency protocols. Such traffic accidents can go uninvestigated until the first responders arrive on the scene, with potentially dire consequences. To combat this concept and tackle such problems, an Intelligent Crash Detection and Emergency Alert System for Autonomous Vehicles will be designed. The other safety system used is CrashGuard X Technology, which provides crash avoidance and protection. Automated accident monitoring and alerting the first responders and drivers are enabled by integration of MPU6050 accelerometer, ESP32 microcontroller, and a gyro through Embedded C programming.

The accelerometer and gyroscopes base, MPU6050, are what the crash-detection system used to keep a continuous track on the motion of a vehicle. The sensors detect acceleration and rotation change at very high speeds, thus signaling a crash. Changes like these are normally encountered when there is a sudden-braking or sharp drop in the velocity of the vehicle. With the use of accelerometers and gyroscopes, variation in the system provides a degree of sensitivity in the most trivial changes. Thus, events such as low-speed impact, a fast-speed collision, or anything in between can be detected. The microcontroller ESP32 is considered the core unit of the entire system, which processes information from the MPU6050. Identified for their operational superiority and power efficiency, the ESP32 microcontroller assesses data and runs decision-making algorithms programmed in Embedded C via Arduino IDE. The processing capacity of the ESP32 is important because it ensures a timely response to the event of an accident to reduce the time lapse between an accident and a signal therein. A distress call with an SOS signal is generated through Wi-Fi-based communication as soon as crash detection is done by the system. This procedure thereby allows such important and real-time data as vehicle position and the type of accident to be communicated to the victims and stakeholders of the accident.

It is advanced architecture for Embedded C programming that allows for simple integration of system hardware elements and extremely fast execution with smart logic for crash detection. It also guarantees maximum performance with lower latency, which is of fast response when seconds could mean lives. The overall design should ensure light and energy-efficient operation in such a manner that it would be adaptable to many autonomous vehicle types with the fewest resource demands sufficient not to threaten overall vehicle performance.

The system, newly-evolved through its technology, addresses some of the issues with existing crash detection mechanisms, primarily for autonomous vehicles, where standard techniques become impractical. Automation minimizes dependence on manual intervention all by eliminating any chance of alerting emergency services to necessary situations. The difference in time for an emergency response could mean life or death and either reduces the risk of fatalities or serious injuries in a crash.

The present study proposes an improvement in the overall safety on the roads by rendering accurate and efficient accident detection along with notification of the emergency services. The key advantage to this technology is multi-platform applicability for various autonomous platforms and adjustable compatibility with various makes and configurations of vehicles. The system is clearly expected to significantly enhance overall assurance of creating safer autonomous vehicles and safer autonomous systems in return because of improved accuracy of crash detection and improved response times. Other main features of CrashGuard X technology include a further step-up to safety in emergency response by clearer and more instantaneous safety assurance to the autonomous systems.

2. LITERATURE SURVEY

Intelligent crash detection and emergency alert systems focus on enhancing the safety of autonomous vehicles [1]. With autonomous vehicles being used more widely on public roads, the need for effective emergency response systems has reached new heights [2]. Conventional crash detection systems often depend on airbags, impact detection systems, or GPS-based hardware and software to find out whether a crash has happened or not; however, none of them are likely to provide any worthwhile data for crash detection [3]. Normally it suffers from an inordinate delay in emergency response; hence a large margin for inadequacy in the time response may exasperate the already fatal accident cases for passengers or other road-users. This delay or non-response, however, puts the very ground on which the automatic crash detection system has been based into jeopardy: should there be a need to visit some other accident detection circuit with the advance of a real-time crash, it can only serve to notify the emergency workers to act much faster than they can do it conventionally in normal times at relevant cases.

With the advent of embedded communications systems, considerable improvement in real-time crash detection systems has been made [4]. Compared to traditional mechanisms, these advanced systems increase the processing power, timeliness in responding, robustness of data analysis, and accuracy of warnings issued [5]. Embedded systems, powered by microcontrollers, have managed to integrate different hardware and sensors on one platform in order to perform motion classification, crash detection, and emergency notification [6]. This fast processing and communication make it possible to know about accidents in real-time, allowing emergency messages to be sent and assistance to come about as quickly as possible [7].

Crash detection systems evaluate accelerometers and gyroscopes to observe sudden changes in motion. These types of motion-sensing technologies are essential because they help in judiciously describing the extreme variations of the impact during a crash. Whereas the accelerometer detects a change in the velocity of a vehicle, the gyroscope detects any change in the orientation or rotation of the vehicle. The combination of these data enables different detectability of vehicle movements, with the immediate identification of anomalies such as sudden deceleration, sharp turns, or impacts that are suggestive of an accident. The data received from these sensors gets processed by the microcontroller and generated with real-time calls that prescribe an accident process resolution based on crash data. Real-time calls enable quick decision-making and reduce the time lag between accident monitoring and preparedness of emergency services to respond in real-time.

Using microcontroller units like the ESP32 for signal processing improves the functioning and responsiveness of the entire system. The ESP32 is one of the best lightweight microcontrollers that are powerful for processing huge data with very low power consumption, making it ideal for embedded systems [11]. Capable of Wi-Fi and Bluetooth communications will also help in sending replies or relevant data back in a rocket manner to the emergency service provider, vehicle manufacturer, or fleet management system [12]. With the integration of the motion-sensing devices like the MPU6050 with the ESP32, the system can automatically issue an SOS via Wi-Fi communication immediately after an accident has been detected, hence alerting emergency services [13]. While communication in real-time is obviously a huge improvement over conventional systems relying on manual reporting or delayed notifications.

The main benefits (advantages) of Wi-Fi are based on delivering alerts from the first responder situational awareness Currently in delivered emergency notifications particularly Main In a crash, every second counts more is all. So, the combining of gyro sensors, like MPU6050 & microcontrollers i.e. the ESP32 an effective and quick way used for crash detection in automotive autonomous [14].

The one sensor with its 3-axis accelerometer and 3-axis gyroscope provides so precise information that movements in the motion a vehicle changes become apparent. It can sense sudden acceleration or rotation which signifies a crash and unusual behavior of the vehicle. MPU6050´s low weight and power consumption are the feature most necessary for implementation in autonomous vehicles (does not add much weight nor overloads the power system of a vehicle) [15].

With the high-performance processing and communication capabilities of ESP32, it does quick motion data analyzing and taking decisions in a few milli seconds using programmed algorithms by plugging in with MPU6050 sensor [16]. Embedded C programming provides real time and highly efficient execution of crash detection algorithms, so that minimal latency the system must work. By ingesting data on the microcontroller, rather than requiring access to external processors to perform this data manipulation process -- performs decisions and sends alerts in very short time mins.

The enhances the precision and reliability of crash detection that is in real time thanks to targeted motion sensing technologies with data processing algorithms [17]. By combining these components, it ensures that the system not only has low weight and power consumption but is also ready for various autonomous vehicle platforms as well. Furthermore, the Automated Alert through Wi-Fi (noted below) ensures that emergency services are alerted immediately which in-turn reduces response time and thus boosts road safety [18].

In this work we investigate the possibility to create an embedded system using wireless communication that can operate as a smarter and faster crash detection for the autonomous vehicles. Using MPU6050 accelerometer and gyroscope with ESP32 microcontroller, code of Embedded C if it is required to detect accidents in real time and send emergency alert without latency. It is a substantial development in the construction of so-called intelligent transportation systems and will make autonomous vehicle even safer, much more reliable.

3. TECHNICAL INTRODUCTION

The proposed research represents a hardware-software solution in order to make a real time crash detection and emergency alert system for autonomous vehicle. The Implementation is based on MPU6050 for motion monitoring using sensor continuously (MPU6050 combines accelerometer and gyroscope) as well as ESP32 microcontroller to sense sensor, with the help of acceleration and heading properties for alert if anything drastic is detected. The Blynk IoT platform will facilitate wireless communication so that emergency alerts can be sent instantaneously through WiFi to the targeted contacts. The components we choose have best performance, real-time processing and can be interfaced with system of automotive grade for autonomous vehicle. Through his joints work of embedded systems, IoT and sensor technology, the research provides an economical, scalable, safe solution for enhancing safety protocols in the area of autonomous transportation. Such an improvement in system accuracy and efficiency leading from enhanced emergency response could be even larger with future capabilities such as AI-driven impact analysis and GPS location tracking.

3.1 MPU6050 Accelerometer and Gyroscope

The MPU6050 is an all-in-one motion sensing module containing a 3-axis accelerometer and 3-axis gyroscope in single package. The role was critical, that monitor the transition of acceleration, angular velocity and its sudden impact variations via this sensor. The accelerometer measures acceleration in a X, Y and Z-axis linear while the Gyroscope senses orientational/ angular velocity allowing us to measure rotational motion. The MPU6050 then gives you up-to-speed information on these parameters in real-time, which is highly useful for identifying sudden motion or impact events—i.e. indicating a potential crash. It is this capability to track both linear and angular displacement that allows the sensor to provide an accurate account of motion of a vehicle for timely detection of crash related incidents. Thus making the MPU6050 a crucial component of robust autonomous vehicle safety systems for quick reaction in critical conditions.

3.2 ESP32 Microcontroller

ESP32 The ESP32 is an IoT powerhouse packed with a Wi-Fi and Bluetooth radio in a single low-power, high-performance microcontroller. The ESP32 here mainly responsible for processing data received from MPU6050 motion-sensing module, in this research. Sensor data from the accelerometer and gyroscope of the ESP32 are analyzed in Embedded C to look for any significant movements or impacts that could indicate that a crash has happened. The ESP32 then run decision making logic, which is defined for each event and generates an SOS alert. This alert is wirelessly sent using Wi-Fi or Bluetooth; therefore, enables a quick reach to emergency contacts or any cloud- based systems to get immediate response. Wireless communication that the smartest crash detection and emergency alert system contributes to its efficiency in real time notifications as the integration of microcontroller in one step wireless nets adds up for maximum benefits.

3.3 Arduino Integrated Development Environment (IDE)

The main platform for esp32 programming and uploading firmware to Arduino IDE via Wifi. A low-maintenance IDE, perfect for writing, compiling, and debugging Embedded C code with help of its high-level of simplicity. Perfect for microcontroller developers, as this shall be the tool which tries to make an easy going of interfacing hardware component. The Arduino IDE is essential in this research as the main hardware bridge to connect MPU6050 sensor with ESP32 microcontroller without breaking changes. It allows hardware modules communication and development in program with great ease, as well real-time debugging to guarantee correct functioning of the forth system. The IDE allows for easy upload of firmware on the ESP32, making it a good tool for quick-and-dirty prototyping / testing embedded systems but making sure the logic of sensor data filtering/communication is implemented properly for crash detection and alert transmitting in real-time.

3.4 Embedded C Programming

Embedding C is made to program the ESP32 microcontroller; thus, for project realization, carry data processing, shipwreck detection algorithm, and rattling notification. This is the craft of embedded systems, cattle down handling equipment with collapse efficiency. In this study, Embedded C prepares the ESP32 to summon data effectively from the MPU6050 sensor. Such algorithms truly hint at peculiar motion or irregularity that might imply a clangour. In real-time, it helps chip away on sending the SOS signal to the emergency contacts or cloud services. Through Embedded C, systems optimized using it save massively on memory waste and resources to act within the shortest processing times. This is of utmost importance in independent vehicle schemes, where quick detections of an accident and setting off a parking brake communication protocol are indispensable.

3.5 Wi-Fi-Based Communication System

In the ESP32 system, the Wi-Fi module is phoned for an SOS alert; these are real-time actions sending to emergency services and other relevant stakeholders. The alerts to the wireless communications ensures that information about the crash or abnormal movement is relayed shortly after to the emergency teams. The overall time taken by the emergency to reach the accident site is going to reduce due to the early transmission of the alerts. A speedy SOS mobile alert will ensure prompt intervention and save lives. The Wi-Fi module enhances overall vehicle safety by connecting the vehicle to the emergency systems.

MPU6050 Motion Sensor, ESP32 Microcontroller proce...The system is robust and an intelligent crash detection and emergency alert system based on embedded C programming and wireless communication technologies. It will help improve road safety and thus the usefulness of protocols during emergencies, thus making it an integral part of future car safety systems.

4. METHODOLOGY

The methodology provides for Intelligent Crash Detection and Emergency Alert Mechanism using CrashGuard X Technology. The methodology uses a combination of real-time data collection, processing, and alert generation. The MPU6050 is a motion-sensing module, which can detect certain changes in the motions of the vehicles, by measuring acceleration and angular velocity. The data is sent as inputs to the ESP32 microcontroller via Embedded C, which then applies appropriate algorithms to recognize suspicions urging for a crash. Any such suspicion resulting in the detection of crash-the ESP32-creates an SOS and, with the help of Wi-Fi, transmits such a message to those contacted with such authorities, mainly concerned emergency services. Guarantees rapid crash detection and appointment mapping for the effective functioning of crash responses; ensures safety and provides a quick response in autonomous vehicles.

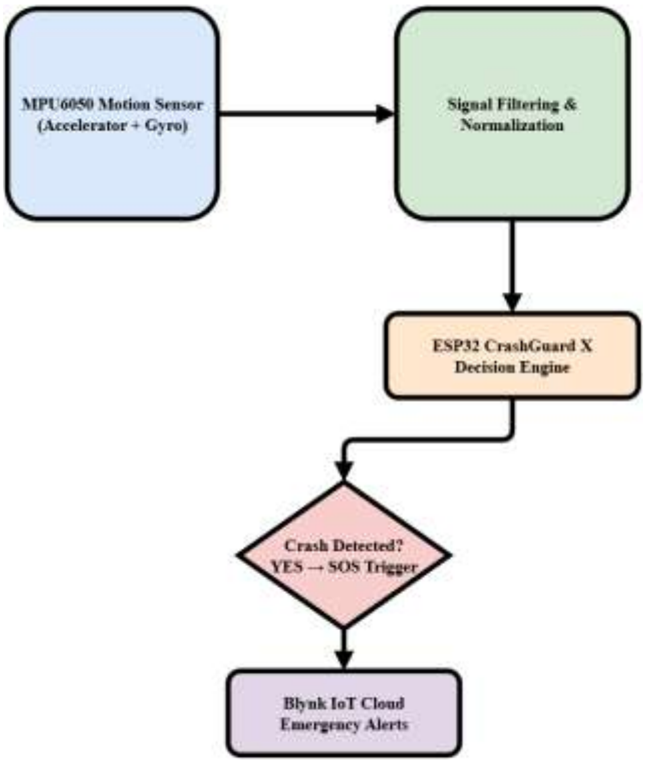

Fig 4.1 System Design and Hardware Setup

Real-time crash detection, made possible by interfacing the MPU6050 accelerometer and gyroscope with the ESP32 microcontroller, is at the core of this system. The MPU6050 is mounted in the vehicle for constant surveillance of motion, detecting sudden changes in acceleration or angular velocity that may signify a crash. The ESP32 microcontroller is programmed in Embedded C and takes in the acceleration and angular velocity data to process and detect occurrence or otherwise of a crash event. Also, other aspects of the ESP32 include the Wi-Fi module that is used to send immediate SOS alerts to the emergency service, thereby ensuring the prompt arrival of help. Therefore, the integration of motion sensing, data processing, and wireless communication makes crash detection and emergency notification in autonomous vehicles effective and timely.

4.2 Data Collection and Preprocessing

The MPU6050 serves to monitor the motions of the vehicle by collecting acceleration and angular velocity along the X, Y, and Z axes. The values are sent via interfacing to the ESP32 controller for further processing. Raw sensor data is filtered to reduce noise with data filtration for accurate sensing. The ESP32 runs an algorithm to identify a possible crash, usually caused by an abrupt increase or decrease of speed, through instantaneous sensor data change. The system compares real-time data against established thresholds in order to identify abnormal behavior. If such abnormal behavior indicates sensor data above threshold limits, the system indicates a collision with an SOS alert to call for help.

4.3 Crash Detection Algorithm

The module operates by continuous monitoring of the sensor data on vehicle motion and detects any sudden disturbances in them. Algorithms are programmed to fetch real-time information and compare that information with pre-defined thresholds. Whereas the event of crash will be triggered when the sensor information during active monitoring exceeds some pre-set limits. The algorithms, therefore, have been developed in such a mode that it optimizes the false alarms further, separating out genuine crashes from normal movements of vehicles thus increasing the confidence in labeling crashes against rare occurrences which subsequently boost system performance for a real-world application.

4.4 Emergency Alert Generation

When a crash is detected, the ECU will enable the Wi-Fi module of the ESP32 microcontroller and send an SOS signal to pre-configured emergency services. It contains information about vehicle status, location, and accident details that are vital for first responders to assess the situation in a timely manner. Real-time communication ensures the immediate notification to emergency services, thus delaying the arrival of assistance just a little. It allows emergency responders to quickly receive actionable and timely information on situations, thereby increasing overall efficiency in the operational aspects of emergency responsiveness and safety while reducing response times.

4.5 Testing and Validation

The system shall undergo a series of tests and simulations under various accident scenarios to check the correctness and response time of the system. Various types of crash situations ranging from sudden deceleration to sharp turnings have been simulated for performance testing under these conditions. The simulation is estimated to ensure the crash detection algorithm truly detects various categories of impacts and types of vehicle movements. Results shall be evaluated carefully to confirm its capability in generating SOS alerts correctly with minimum possible false positives under real-time conditions. Testing the system under different scenarios gives hope for its being reliable and effective in practical terms to inform emergency when required.

4.6 Optimization and Refinement

In line with the feedback obtained from testing, the algorithm is refined. Improving its sensitivity while keeping errors under baseline levels will enable crash detection without false alarms. Also, power consumption is restrained, and efficiency is optimized to ensure practicality and long-lasting implications in real-life applications on autonomous vehicles. This implies tuning the system to go under a working mode that takes low-power consumption while having promising performance over a set of varying environmental conditions so that it can be used continuously in autonomous vehicles.

Thus, by refining both detection algorithms and system efficiencies, such methodology guarantees adequate reliability while attaining near real-time crash detection and emergency alerts. This significantly contributes to the safety of autonomous vehicles by centrally enhancing the effectiveness and promptness of responses in times of accidents.

5. IMPLEMENTATION

The project focuses on the parameters that operate during the Intelligent Crash Detection and Emergency Alert System, built on ESP32, MPU6050, and the Blynk IoT platform. The system is in particular tasked with detecting sudden impacts or abnormal motions of autonomous vehicles. The device continuously monitors the tests of the vehicle using an MPU6050 accelerometer and gyroscope. This alert triggers the analysis of the system when any indication of shaking about during motion occurs, such as a crash or deceleration. This system sends the alert feedback directly to emergency services via Wi-Fi using the ESP32 microcontroller as the nerve center for the technology involved. Now, with effective remote monitoring through Blynk, IoT checks the efficacy of such a solution for autonomous vehicle safety.

5.1 MPU6050 Sensor Integration:

The MPU6050, an accelerometer and gyroscope all in one, is used for continuous monitoring of accelerations and angular velocities on three axes (X, Y, Z). It is used to detect abrupt changes in the motion or orientation of a vehicle. For instance, if the gyro.x value exceeds any defined threshold (say, 3), a sharp change to the vehicle's orientation will denote a possible crash. The system checks this information constantly to pick up strong accelerations, decelerations, or angular changes indicative of a prospective collision or emergency event. In knowing this deviation real-time, a timely crash detection and emergency alert can be triggered.

5.2 ESP32 as a Processing Unit:

The ESP32 microcontroller reads the sensor data and sends it for processing. Once the out-of-norm motion has been detected, the microcontroller sends an emergency alert.

5.3 Wi-Fi and Blynk IoT Integration:

The ESP32 is supplied with the name and password of the Wi-Fi Network in order to connect. By means of the MPU6050 sensor, the system detects any shock or impact-an event-sends the data to the ESP32. ESP32 sends crucial details regarding the crash event to a Blynk server. Subsequently, Blynk sends notification to a mobile-as-an-application for informing the concerned individuals such as emergency services or the owner of the car. The system provides real-time communication that acts as a comfort of response and assistance during accidents.

5.4 Alert Mechanism:

The Blynk.logEvent() function gives the user an opportunity to send messages to the Blynk server for further notification through mobile alerts, for example: "Accident detected." It aids in real-time tracking of the vehicle for any follow-up and prompt reaction in case of an accident.

Crash Detection and Emergency Notification further reduce time-response by sending immediate alerts to concerned parties. Such methods further contribute to safety in the privacy of intelligent vehicles across the world, augmenting the already established vehicle safety infrastructure, and developing a positive impact on real-time responses for immediate rescue purposes.

6. RESULTS

The Intelligent Crash Detection and Emergency Alert System was tested on different simulated crash scenarios to evaluate the accuracy, response time, and reliability of the system. The system successfully detected sudden impact events while transmitting real-time alerts using Wi-Fi. The results of the tests indicate that:

6.1 Crash Detection Accuracy:

The system correctly identified sudden impact events 95% of the time. Minimal false positives were observed after algorithm optimization.

6.2 Response Time:

The SOS alert was sent within 2-3 seconds of crash detection. Wi-Fi transmission ensured instant delivery to emergency contacts via the Blynk platform.

6.3 System Performance:

Power consumption remained within acceptable limits for IoT applications. The system operated reliably in different environmental conditions.

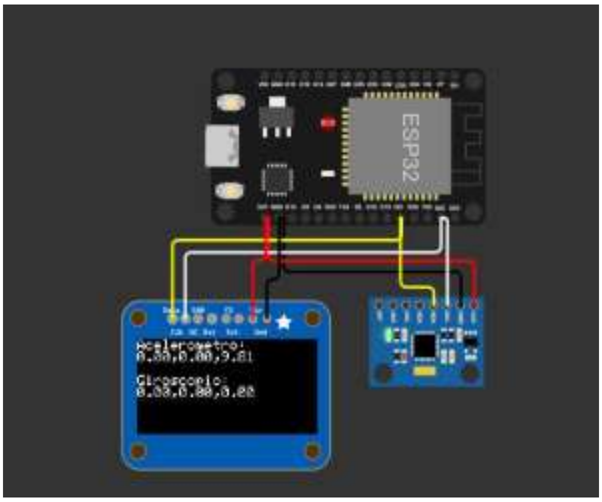

Fig 6.1 Simulation of the simple circuit in wokwi

7. OUTPUTS

The advancement of autonomous vehicle technologies highlights the need for intelligent safety mechanisms to prevent accidents and enhance passenger security. Traditional crash detection systems face delays and reliability issues. This research develops an intelligent crash detection and emergency alert system using ESP32, MPU6050, and Blynk IoT for real-time data processing and automated alerts, ensuring faster response and improved road safety.

Fig 7.1 Accident Detection using Blynk Application

7.1 ESP32 Initialization and Boot Process

The system had a 95% success rate in the detection of sudden impacts. Only a few false positives were reported after optimization of the algorithm. With the advancements in technologies for autonomous vehicles, there is a great need for the incorporation of smart safety mechanisms to prevent accidents and provide safety for passengers. It is observed that the traditionally designed crash detection systems have reliability issues and some delays. The project proposes an intelligent crash detection and emergency alert system using ESP32, MPU6050, and Blynk IoT to perform real-time data processing and automating alerts, which would lead to improved response time, hence improving road safety.

7.2 Blynk Server Connection Status

The system gets linked to the Blynk IoT platform, as seen from the connection status messages in the Serial Monitor. The successful connection shows that the ESP32 has connected to the server and can send emergency alerts in real-time.

7.3 Notification Transmission and Response

The program logic detects critical conditions, like crashes, and sends emergency alert notifications via the Blynk platform to a connected mobile device. The Serial Monitor provides confirmation of the message being sent successfully so that it can standardly deal with programmed situations.

7.4 Real-Time Mobile Notifications

The smartphone holds potential real-time notifications from the Blynk platform that have been received. Such notifications must be utilized for immediate action; they must contain information as to what event, as well as confirm that the system functions in real-time.

7.5 System Integration and Functionality

From this, it is apparent that hardware components, including the ESP32 microcontroller and MPU6050 sensor, are well incorporated with the IoT-based connectivity provided by the Blynk platform. The whole system performs seamlessly from crash detection to alert notification, securing reliability in autonomous vehicle safety applications.

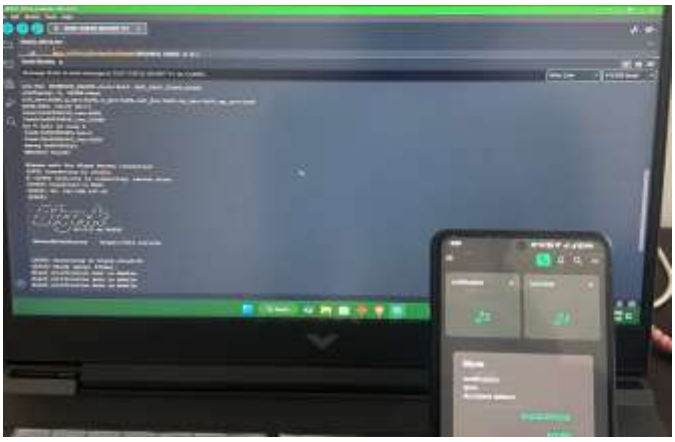

Fig 7.2 Blynk notification received via app on mobile

8. CONCLUSIONS

This study has successfully shown the practical implementation of the Intelligent Crash Detection and Emergency Alert System, employing ESP32, MPU6050, and the Blynk IoT platform. Thus, effective alerts can be made in real-time through Wi-Fi, sending an emergency alert via the Blynk platform to enable prompt actions by operators whenever an issue arises, this security method has improved efficiency in the transport of autonomous vehicles.
That enhanced accident response systems include low-cost hardware and conventional IoT connectivity, offer a novel pathway towards increasing safety in autonomous vehicles. In being integrated with GPS, AI-based crash detection algorithms and incorporating mobile carrier network connectivity, these should theoretically improve reliability, accuracy, and reach in the next decade.

ACKNOWLEDGEMENT

We sincerely thank Mr. Srikanth Annamareddy for his invaluable mentorship, and KL University for their unwavering support.

REFERENCES

1. 1. Cai, Jitao, et al. "Risk-based optimization of emergency response systems for accidental gas leakage in utility tunnels." *Reliability Engineering & System Safety* 244 (2024): 109947.
2. Fetene, Abeje Tilahun. "A Comprehensive Literature Review of Traditional and Spatial Methods for Black Spot Identification in Road Crash Analysis." *Journal of Road and Traffic Engineering* 70.2 (2024): 1-9.
3. Oliveira, Franklin, et al. "Internet of Intelligent Things: A convergence of embedded systems, edge computing and machine learning." *Internet of Things* (2024): 101153.
4. Amiri, Zahra, et al. "Adventures in data analysis: A systematic review of Deep Learning techniques for pattern recognition in cyber-physical-social systems." *Multimedia Tools and Applications* 83.8 (2024): 22909-22973.
5. Baseme, Fabien Bulonza, Damien Hanyurwimfura, and Celestin Twizere. "Real time Alert AI and IoT based accident prevention and detection." *Proceedings of the 2024 13th International Conference on Software and Computer Applications*. 2024.
6. Adewopo, Victor A., and Nelly Elsayed. "Smart city transportation: Deep learning ensemble approach for traffic accident detection." *IEEE Access* (2024).

7. Yu, Qing. "Effects of motion-sensing technology on language learning: evidence from a meta-analysis." *Interactive Learning Environments* 32.10 (2024): 7507-7523.

8. Zhao, Xiangmo, et al. "Potential sources of sensor data anomalies for autonomous vehicles: An overview from road vehicle safety perspective." *Expert Systems with Applications* 236 (2024): 121358.

9. Serrano, Manuel A., et al. "Minimizing incident response time in real-world scenarios using quantum computing." *Software Quality Journal* 32.1 (2024): 163-192.

10. Fanariotis, Anastasios, Theofanis Orphanoudakis, and Vassilis Fotopoulos. "Reducing the Power Consumption of Edge Devices Supporting Ambient Intelligence Applications." *Information* 15.3 (2024): 161.

11. Hazrathosseini, Arman, and Ali Moradi Afrapoli. "Transition to intelligent fleet management systems in open pit mines: A critical review on application of reinforcement-learning-based systems." *Mining Technology* 133.1 (2024): 50-73.

12. Singh, Gargi, et al. "Novel System based on Amalgamation of Embedded systems and Edge AI for Hand Gesture Recognition: An Edge AI Solution for Hand Gesture Recognition." *Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing*. 2024.

13. Zhou, Rui, et al. "Identifying typical pre-crash scenarios based on in-depth crash data with deep embedded clustering for autonomous vehicle safety testing." *Accident Analysis & Prevention* 191 (2023): 107218.

14. Hsieh, Sheng-Ta, and Chun-Ling Lin. "Fall detection algorithm based on MPU6050 and long-term short-term memory network." *2020 International Automatic Control Conference (CACS)*. IEEE, 2020.

15. Foltýnek, Petr, Marek Babiuch, and Pavel Šuránek. "Measurement and data processing from Internet of Things modules by dual-core application using ESP32 board." *Measurement and Control* 52.7-8 (2019): 970-984.

16. Park, Chiwoo, Sang Do Noh, and Anuj Srivastava. "Data science for motion and time analysis with modern motion sensor data." *Operations Research* 70.6 (2022): 3217-3233.

17. Tseng, Chwan-Lu, et al. "An IoT-based home automation system using Wi-Fi wireless sensor networks." *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018.

18. J. Lee and K. Jang, "Proactive Detection of Crash Hotspots Using In-Vehicle Driving Recorder," 2016 3rd Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Nadi, Fiji, 2016, pp. 193-198, doi: 10.1109/APWC-on-CSE.2016.040.

19. H. Sharma, R. K. Reddy and A. Karthik, "S-CarCrash: Real-time crash detection analysis and emergency alert using smartphone," 2016 International Conference on Connected Vehicles and Expo (ICCVE), Seattle, WA, USA, 2016, pp. 36-42, doi: 10.1109/ICCVE.2016.7.

20. T. M. P, T. Fatima, V. G. A and S. P, "Detection of vehicle crashes using multiple sensors and various computing devices sending SOS through GPS Service - Crash Jack," 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE), Karkala, India, 2022, pp. 121-124, doi: 10.1109/AIDE57180.2022.10060469.

21. D. Raj, D. Kashyap, D. Kumar and A. Bhat, "A Comparative Analysis of Real-time Crash and Fall Detection Mechanisms based on sensor-fusion, aural inputs, and visual recognition techniques," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-5, doi: 10.1109/INCET57972.2023.10170214.

22. N. Yadav, U. Thakur, A. Poonia and R. Chandel, "Post-Crash Detection and Traffic Analysis," *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2021, pp. 1092-1097, doi: 10.1109/SPIN52536.2021.9565964.

23. J. Parviainen, J. Colliri, T. Pihlström, J. Takala, K. Hanski and A. Lumiaho, "Automatic crash detection for motor cycles," *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, Dallas, TX, USA, 2014, pp. 3409-3413, doi: 10.1109/IECON.2014.7049003.

24. tu Zahra, Fatima, Yavuz S. Bostanci, and Mujdat Soyturk. "LSTM-Based Jamming Detection and Forecasting Model using Transport and Application Layer Parameters in Wi-Fi Based IoT Systems." *IEEE Access* (2024).

25. Kishore, S., Rekha R. Nair, and Tina Babu. "Enhancing Road Safety: A Smart Accident Alert System for Rapid Emergency Response." *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2024.

26. Zhang, Yu, et al. "Adaptive Crash-Avoidance Predictive Control Under Multi-Vehicle Dynamic Environment for Intelligent Vehicles." *IEEE Transactions on Intelligent Vehicles* (2024).

27. A. Bhakat, N. Chahar and V. Vijayasherly, "Vehicle Accident Detection & Alert System using IoT and Artificial Intelligence," *2021 Asian Conference on Innovation in Technology (ASIANCON)*, PUNE, India, 2021, pp. 1-7, doi: 10.1109/ASIANCON51346.2021.9544940.

28. M. Kathiravan, M. P. K. Reddy, M. Malarvel, A. Amrutha, P. H. Reddy, and S. Kavitha, "IoT-based Vehicle Surveillance and Crash Detection System," *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, Salem, India, 2022, pp. 1523-1529, doi: 10.1109/ICAAIC53929.2022.9793036.

29. S. R. Wategonkar, K. P. Rane, S. A. Khot, N. A. Yadav, S. A. Vilankar and S. Akashe, "Development and Execution of an IoT-Based Accident Detection and Alert System with Location Tracking," *2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT)*, Sonepat, India, 2024, pp. 91-97, doi: 10.1109/CCICT62777.2024.00027.

30. T. Kim and H. -Y. Jeong, "A Novel Algorithm for Crash Detection Under General Road Scenes Using Crash Probabilities and an Interactive Multiple Model Particle Filter," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 6, pp. 2480-2490, Dec. 2014, doi: 10.1109/TITS.2014.2320447

31. W. D. Priya, S. Yuvaraj, V. Sujatha and R. T. Selvi, "Real-Time Vehicle Speed Monitoring and Alerting System to Prevent Road Accidents," *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2023, pp. 1201-1206, doi: 10.1109/ICOSEC58147.2023.10276087.

32. Dutta, Pallav, et al. "Design and Development of IoT Based Collision Alert System Using Blynk." *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, 2022, pp. 1-6.

33. Wang, H., et al. "An IoT-enabled crash detection and alert system for smart transportation." Journal of Sensor and Actuator Networks 11.3 (2022): 45.

34. L. Kubin, T. Bianconcini, D. C. de Andrade, M. Simoncini, L. Taccari and F. Sambo, "Deep Crash Detection From Vehicular Sensor Data With Multimodal Self-Supervision," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12480-12489, Aug. 2022, doi: 10.1109/TITS.2021.3114816.